do by me

简单思路描述

针对这个题，我自己最常规的思路就是直接通过两个循环进行枚举，找到两个数据的和等于target的下标就完事儿了。

代码

```cpp
class Solution {

public:
    vector<int> twoSum(vector<int>& nums, int target) {

        int N = nums.size();

        vector<int> result;

        for (int i = 0; i < N; i++) {

            for (int j = 0; j < N; j++) {

                if (nums[i] + nums[j] == target && i != j) {

                    result.push_back(i);

                    result.push_back(j);

                    break;
                }
            }

            if (result.size() == 2) {

                break;
            }
        }

```

```
        return result;
    }
};
```

Do by me（升级版）

```
class Solution {
public:
    vector<int> twoSum(vector<int>& nums, int target) {

        vector<int> result;
        // 哈希表解决这个问题

        // 哈希表的创建
        unordered_map<int, int> hashtable;

        // 遍历数组
        for (int i = 0; i < nums.size(); i++) {

            int b = target - nums[i];

            // 只需要看遍历数组前面有没有b就可以。（这个比较关键，想想为什么？）

            // 显式定义返回值类型
            unordered_map<int, int>::iterator it = hashtable.find(b);

            // 如果找到了
            if (it != hashtable.end()) {
                result.push_back(it->second);
                result.push_back(i);
                return result;
            }
            // 动态维护哈希表
            hashtable[nums[i]] = i;
        }
        return {};
    }
};
```

官方题解

主要用了两种方法，一种是枚举，和我的思路一样，另一种的哈希表。

1. 两数之和 - 力扣（LeetCode）

知识补充

哈希表模版： Hash哈希模板_哈希模版-CSDN博客